



Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano-Bicocca



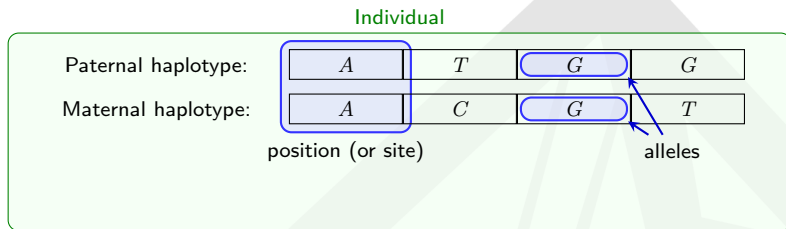
Pure Parsimony Xor Haplotyping

ISBRA 2009

Paola Bonizzoni, Gianluca Della Vedova, Riccardo Dondi,
Yuri Pirola, Romeo Rizzi

`pirola@disco.unimib.it`

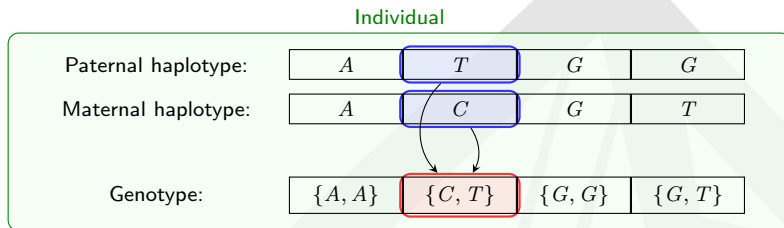
Haplotype inference



Haplotype Inference problem: given a population, to infer haplotypes of individuals from genotypic data.

Why? Haplotypes are valuable but **more expensive** than genotypes.

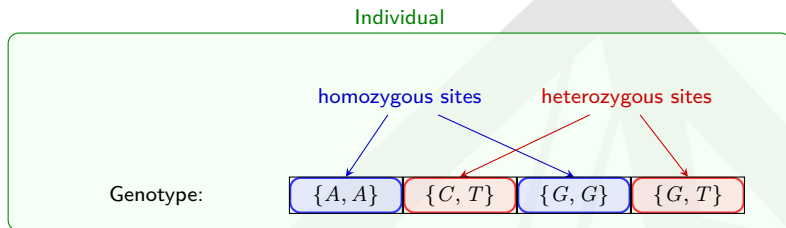
Haplotype inference



Haplotype Inference problem: given a population, to infer haplotypes of individuals from genotypic data.

Why? Haplotypes are valuable but **more expensive** than genotypes.

Haplotype inference



Haplotype Inference problem: given a population, to infer haplotypes of individuals from genotypic data.

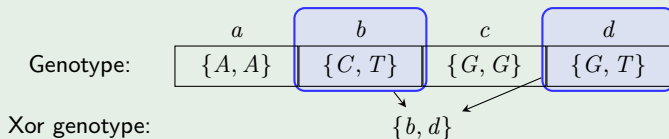
Why? Haplotypes are valuable but **more expensive** than genotypes.

Xor genotypes

Xor genotype: the set of the heterozygous sites (=characters) of the individual.

- may be cheaper to obtain than regular genotypes
- Perfect Phylogeny [Shamir *et al.*, CPM04, IEEE TCBB 08]
- Pure Parsimony: ILP formulation [Brown and Harrower, IEEE TCBB 06]

Example



Results

Pure Parsimony Xor Haplotyping (PPXH):

- Combinatorial properties of the solutions
- Polynomial (exact) algorithms for constrained cases
- Fixed-parameter algorithm
- (*Approximation algorithm*)
- Heuristic algorithm (and experimental analysis)

Problem definition

Definition (PPXH problem)

Given a set X of xor genotypes over a set Σ of characters, find a smallest set H of haplotypes such that for each $x \in X$ there exists a pair of haplotypes h_i, h_j in H : $h_i \oplus h_j = x$.

Example

Xor genotypes

$\{a\}$

$\{b\}$

$\{c\}$

$\{a, b\}$

$\{b, c\}$

Haplotypes

$\{b\}$

$\{a, b\}$

$\{a\}$

$\{a, b, c\}$

Problem definition

Definition (PPXH problem)

Given a set X of xor genotypes over a set Σ of characters, find a smallest set H of haplotypes such that for each $x \in X$ there exists a pair of haplotypes h_i, h_j in H : $h_i \oplus h_j = x$.

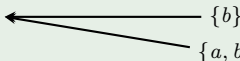
Example

Xor genotypes

$\{a\}$
 $\{b\}$
 $\{c\}$
 $\{a, b\}$
 $\{b, c\}$

Haplotypes

$\{b\}$
 $\{a, b\}$
 $\{a\}$
 $\{a, b, c\}$

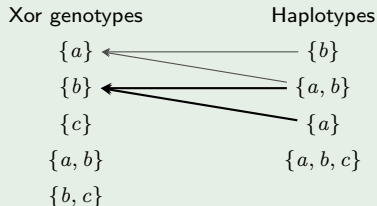


Problem definition

Definition (PPXH problem)

Given a set X of xor genotypes over a set Σ of characters, find a smallest set H of haplotypes such that for each $x \in X$ there exists a pair of haplotypes h_i, h_j in H : $h_i \oplus h_j = x$.

Example

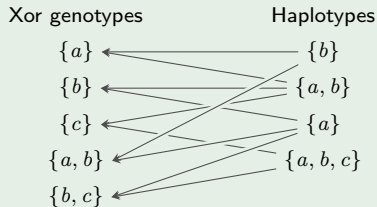


Problem definition

Definition (PPXH problem)

Given a set X of xor genotypes over a set Σ of characters, find a smallest set H of haplotypes such that for each $x \in X$ there exists a pair of haplotypes h_i, h_j in H : $h_i \oplus h_j = x$.

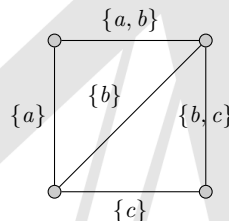
Example



Xor-graph - Graph representation

Xor-graph:

- vertex = haplotype
- edge = xor genotype



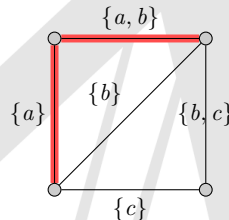
Properties

- character \Rightarrow cut
- cycle $\Rightarrow \text{XOR}(\text{edges}) = \emptyset$

Xor-graph - Graph representation

Xor-graph:

- vertex = haplotype
- edge = xor genotype



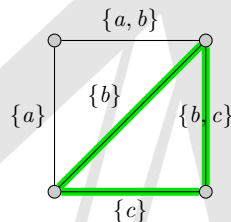
Properties

- character \Rightarrow cut example: a
- cycle $\Rightarrow \text{XOR}(\text{edges}) = \emptyset$

Xor-graph - Graph representation

Xor-graph:

- vertex = haplotype
- edge = xor genotype



Properties

- character \Rightarrow cut
- cycle $\Rightarrow \text{XOR}(\text{edges}) = \emptyset$ example: $\{b\}, \{b, c\}, \{c\}$

Matrix representation

Example

Xor genotypes = $\{\{a, d\}, \{b\}, \{c, d\}, \{a, b, c\}\}$

X	a	b	c	d
x_1	1	0	0	1
x_2	0	1	0	0
x_3	0	0	1	1
x_4	1	1	1	0

Matrix representation

Example

Xor genotypes = $\{\{a, d\}, \{b\}, \{c, d\}, \{a, b, c\}\}$

X	a	b	c	d
x_1	1	0	0	1
x_2	0	1	0	0
x_3	0	0	1	1
x_4	1	1	1	0

Matrix representation

Example

Xor genotypes = $\{\{a, d\}, \{b\}, \{c, d\}, \{a, b, c\}\}$

Haplotypes = $\{\{a, c\}, \{c, d\}, \{b, c, d\}, \{b\}\}$

X	a	b	c	d
x_1	1	0	0	1
x_2	0	1	0	0
x_3	0	0	1	1
x_4	1	1	1	0

H	a	b	c	d
h_1	1	0	1	0
h_2	0	0	1	1
h_3	0	1	1	1
h_4	0	1	0	0

Matrix representation

Example

Xor genotypes = $\{\{a, d\}, \{b\}, \{c, d\}, \{a, b, c\}\}$

Haplotypes = $\{\{a, c\}, \{c, d\}, \{b, c, d\}, \{b\}\}$

X	a	b	c	d
x_1	1	0	0	1
x_2	0	1	0	0
x_3	0	0	1	1
x_4	1	1	1	0

H	a	b	c	d
h_1	1	0	1	0
h_2	0	0	1	1
h_3	0	1	1	1
h_4	0	1	0	0

Matrix representation

Example

Xor genotypes = $\{\{a, d\}, \{b\}, \{c, d\}, \{a, b, c\}\}$

Haplotypes = $\{\{a, c\}, \{c, d\}, \{b, c, d\}, \{b\}\}$

X	a	b	c	d		H	a	b	c	d
x_1	1	0	0	1	← \oplus	h_1	1	0	1	0
x_2	0	1	0	0		h_2	0	0	1	1
x_3	0	0	1	1		h_3	0	1	1	1
x_4	1	1	1	0		h_4	0	1	0	0

Matrix representation


X	a	b	c	d
x_1	1	0	0	1
x_2	0	1	0	0
x_3	0	0	1	1
x_4	1	1	1	0

H	a	b	c	d
h_1	1	0	1	0
h_2	0	0	1	1
h_3	0	1	1	1
h_4	0	1	0	0

Observations

Matrix representation

X	a	b	c	d
x_1	1	0	0	1
x_2	0	1	0	0
x_3	0	0	1	1
x_4	1	1	1	0

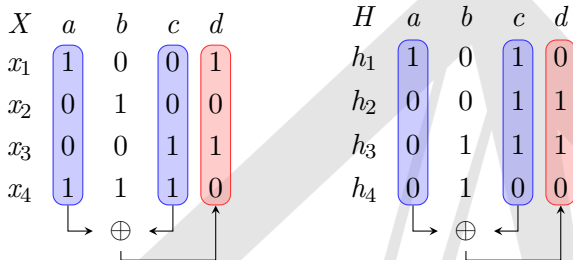


H	a	b	c	d
h_1	1	0	1	0
h_2	0	0	1	1
h_3	0	1	1	1
h_4	0	1	0	0

Observations

On X , $a \oplus c = d$

Matrix representation



Observations

On X , $a \oplus c = d$

\Rightarrow exists H , $a \oplus c = d$

Matrix representation

X	a	b	c	d
x_1	1	0	0	1
x_2	0	1	0	0
x_3	0	0	1	1
x_4	1	1	1	0

H	a	b	c	d
h_1	1	0	1	0
h_2	0	0	1	1
h_3	0	1	1	1
h_4	0	1	0	0

Diagram illustrating the relationship between matrices X and H . In matrix X , columns a and c are highlighted in blue, and column d is highlighted in red. In matrix H , columns a and c are highlighted in blue, and column d is highlighted in red. Arrows indicate that the XOR operation (\oplus) is applied to columns a and c to produce column d in both matrices.

Observations

On X , $a \oplus c = d$

\Rightarrow exists H , $a \oplus c = d$

\Rightarrow **d can be removed!**

Reduction of the instance

Definition (Reduced genotype matrix)

Reduced genotype matrix \Rightarrow no more columns can be removed

Efficient reduction process via the Gauss elimination algorithm.

Lemma (Lower bound)

Let X be a *reduced* genotype matrix with m characters, then the optimal solution has, at least, $m + 1$ haplotypes.

Polytime algorithm for PPXH(2, ∞)

Input: A reduced genotype matrix X over the character set Σ with at most 2 characters in each genotype.

Output: The set of haplotypes $H := \{\emptyset\} \cup \{\{c_i\} \mid c_i \in \Sigma\}$.

Correctness:

- H is a solution.
- H is **optimal** because $|H| = |\Sigma| + 1$ (it meets the lower bound).

Polytime algorithm for $\text{PPXH}(\infty, 2)$

Lemma

If every minimal zero-sum subset of genotypes labels a simple cycle of a xor-graph, then the xor-graph is optimal.

Example

X	a	b	c	d	e
x_1	1	1	1	0	0
x_2	0	1	0	0	0
x_3	1	0	0	0	0
x_4	0	0	1	0	0
x_5	0	0	0	1	1
x_6	0	0	0	1	0
x_7	0	0	0	0	1

Polytime algorithm for $\text{PPXH}(\infty, 2)$

Lemma

If every minimal zero-sum subset of genotypes labels a simple cycle of a xor-graph, then the xor-graph is optimal.

Example

X	a	b	c	d	e
x_1	1	1	1	0	0
x_2	0	1	0	0	0
x_3	1	0	0	0	0
x_4	0	0	1	0	0
x_5	0	0	0	1	1
x_6	0	0	0	1	0
x_7	0	0	0	0	1

Polytime algorithm for $\text{PPXH}(\infty, 2)$

Lemma

If every minimal zero-sum subset of genotypes labels a simple cycle of a xor-graph, then the xor-graph is optimal.

Example

X	a	b	c	d	e
x_1	1	1	1	0	0
x_2	0	1	0	0	0
x_3	1	0	0	0	0
x_4	0	0	1	0	0
x_5	0	0	0	1	1
x_6	0	0	0	1	0
x_7	0	0	0	0	1

Polytime algorithm for $\text{PPXH}(\infty, 2)$

Lemma

If every minimal zero-sum subset of genotypes labels a simple cycle of a xor-graph, then the xor-graph is optimal.

Example

X	a	b	c	d	e
x_1	1	1	1	0	0
x_2	0	1	0	0	0
x_3	1	0	0	0	0
x_4	0	0	1	0	0
x_5	0	0	0	1	1
x_6	0	0	0	1	0
x_7	0	0	0	0	1

Polytime algorithm for $\text{PPXH}(\infty, 2)$

Lemma

If every minimal zero-sum subset of genotypes labels a simple cycle of a xor-graph, then the xor-graph is optimal.

Example

X	a	b	c	d	e
x_1	1	1	1	0	0
x_2	0	1	0	0	0
x_3	1	0	0	0	0
x_4	0	0	1	0	0
x_5	0	0	0	1	1
x_6	0	0	0	1	0
x_7	0	0	0	0	1

Polytime algorithm for $\text{PPXH}(\infty, 2)$

Lemma

If every minimal zero-sum subset of genotypes labels a simple cycle of a xor-graph, then the xor-graph is optimal.

Example

X	a	b	c	d	e
x_1	1	1	1	0	0
x_2	0	1	0	0	0
x_3	1	0	0	0	0
x_4	0	0	1	0	0
x_5	0	0	0	1	1
x_6	0	0	0	1	0
x_7	0	0	0	0	1

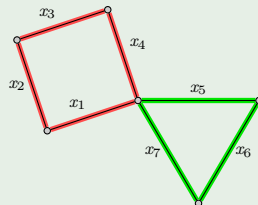
Polytime algorithm for $\text{PPXH}(\infty, 2)$

Lemma

If every minimal zero-sum subset of genotypes labels a simple cycle of a xor-graph, then the xor-graph is optimal.

Example

X	a	b	c	d	e
x_1	1	1	1	0	0
x_2	0	1	0	0	0
x_3	1	0	0	0	0
x_4	0	0	1	0	0
x_5	0	0	0	1	1
x_6	0	0	0	1	0
x_7	0	0	0	0	1



Fixed-parameter algorithm

Lemma (Lower bound) - recall

Let X be a $n \times m$ reduced genotype matrix, and H an optimal solution for X with k distinct haplotypes. Then $m \leq k$.

H has $k \times m \leq k \times k = k^2$ entries
 $\Rightarrow 2^{k^2}$ possible solutions.

Exhaustive enumeration is feasible if k is small.

- naive exhaustive enumeration $\Rightarrow O(2^{k^2} k^2 nm)$
- using Gray codes and Binary tries $\Rightarrow O(nm + 2^{k^2} km)$

Heuristic algorithm

Basic idea: zero-sum subsets of genotypes \Rightarrow cycles.

Rationale: more cycles = less vertices (haplotypes).

Algorithm sketch:

- obtain zero-sum subsets via *Gauss elimination*
- build a graph via *Graph realization* [Bixby and Wagner, 1988]

Complexity: $O(\alpha(n, m)n^3m)$, where $\alpha(n, m) \leq 5$ for any “reasonable” value of m and n .

Experimental analysis

On synthetic genotypes:

- **Data:** a set of m genotypes obtained by extracting random pairs from a set of h random haplotypes.
- **Results:** the ratio between the number of haplotypes obtained and h always **less than 1.58** (for instance sizes between $m = 100$ and $h = 50$, and $m = 400$ and $h = 266$).

On real genotypes:

- **Data:** Phase I dataset of HapMap project (various populations, from 44 genotypes and 184604 sites to 90 genotypes and 91812 sites).
- **Results:** running time always **less than 5 seconds**.

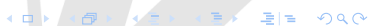
Conclusions and open problems

Results:

- Analysis of **combinatorial properties** of solution representations
- Design of **exact algorithms** under specific assumptions
 - polytime algorithms for constrained PPXH
 - fixed-parameter algorithm for instances with “small” solutions
- (Design of an **approximation algorithm**)
- Design of an effective **heuristic algorithm**
 - good approximation ratio on synthetic data
 - fast on real large instances

Open problems:

- Computational complexity
- Mixing genotype and haplotype data to improve accuracy



Genetic models & methods

Additional assumptions are needed to resolve ambiguity.

Common approaches:

- Statistical methods \Rightarrow ML
- Combinatorial methods:
 - Perfect Phylogeny
 - Pure Parsimony
- (Pedigrees)

Zero-sum property

Lemma (Xor-graph)

A labelled graph represents a solution iff the xor-sum of the edge labels (= xor genotypes) of every cycle is the empty set.

Proof: let $C = \langle h_1, \dots, h_n = h_1 \rangle$ be a cycle, each pair (h_i, h_{i+1}) of consecutive haplotypes solves a genotype x_i . Then

$$x_1 \oplus \dots \oplus x_{n-1} = (h_1 \oplus h_2) \oplus (h_2 \oplus h_3) \oplus \dots \oplus (h_{n-1} \oplus h_1) = \emptyset$$
 since $h_i \oplus h_i = \emptyset$. □

The PPXH problem can be reformulated as the problem to build the smallest Xor-graph.

Reduction of the instance

Definition

A genotype matrix X is *reduced* iff its column vectors are linearly independent.

The reduction process can be efficiently performed by the Gauss elimination algorithm.

Lemma

Let X be a reduced genotype matrix with m characters, then the optimal solution has, at least, $m + 1$ haplotypes.

Proof: the m characters induce m linearly independent cuts in a xor-graph. A graph with m independent cuts has at least $m + 1$ vertices (=haplotypes).

Constrained PPXH

Definition (Constrained PPXH)

PPXH(p, q):

- p , max length of a genotype
- q , max number of genotypes sharing the same character

Example:

X	a	b	c	d
x_1	1	1	1	0
x_2	1	0	0	1
x_3	0	1	1	0
x_4	0	1	0	1

- $p = 3$ because genotype x_1 contains 3 characters
- $q = 3$ because character b belongs to 3 genotypes

Fixed-parameter algorithm

Lemma

Let X be a $n \times m$ reduced genotype matrix, and H an optimal solution for X with k distinct haplotypes. Then $m \leq k$.

Proof:

Since we have m linearly independent columns, there are also m linearly independent rows (=genotypes).

Having $k < m$ haplotypes that explain all genotypes contradicts the linear independence of m genotypes. □

Heuristic algorithm

Algorithm **PPXH**(X)

- ① Find a collection C of zero-sum subsets of X
- ② $C' \leftarrow \emptyset$
- ③ **While** $C \neq \emptyset$ **do**
 - pick a random zero-sum subset c from C
 - **if** $C' \cup \{c\}$ admits Graph Realization \mathcal{G}'
then continue
else remove from X the genotypes which label the last Graph Realization \mathcal{G}' , and start from step 2
- ④ Terminate when $X = \emptyset$

Heuristic algorithm - Example

X	a	b	c
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	1	1	0
x_5	0	1	1
x_6	1	0	1
x_7	1	1	1

Heuristic algorithm - Example

X	a	b	c
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	1	1	0
x_5	0	1	1
x_6	1	0	1
x_7	1	1	1

Zero sum subsets

x_1, x_2, x_4

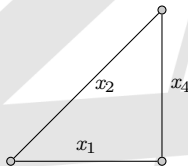
x_2, x_3, x_5

x_1, x_3, x_6

x_1, x_2, x_3, x_7

Heuristic algorithm - Example

X	a	b	c
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	1	1	0
x_5	0	1	1
x_6	1	0	1
x_7	1	1	1



Zero sum subsets

x_1, x_2, x_4

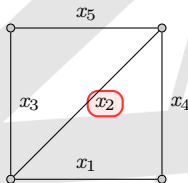
x_2, x_3, x_5

x_1, x_3, x_6

x_1, x_2, x_3, x_7

Heuristic algorithm - Example

X	a	b	c
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	1	1	0
x_5	0	1	1
x_6	1	0	1
x_7	1	1	1



Zero sum subsets

x_1, x_2, x_4

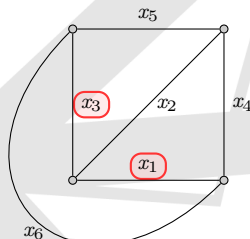
x_2, x_3, x_5

x_1, x_3, x_6

x_1, x_2, x_3, x_7

Heuristic algorithm - Example

X	a	b	c
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	1	1	0
x_5	0	1	1
x_6	1	0	1
x_7	1	1	1



Zero sum subsets

x_1, x_2, x_4

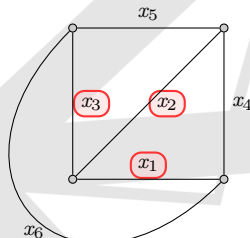
x_2, x_3, x_5

x_1, x_3, x_6

x_1, x_2, x_3, x_7

Heuristic algorithm - Example

X	a	b	c
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	1	1	0
x_5	0	1	1
x_6	1	0	1
x_7	1	1	1



Zero sum subsets

x_1, x_2, x_4

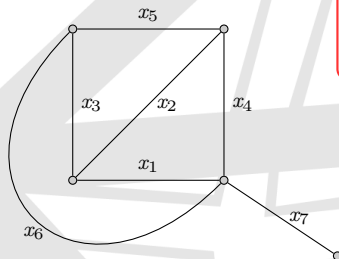
x_2, x_3, x_5

x_1, x_3, x_6

x_1, x_2, x_3, x_7

Heuristic algorithm - Example

X	a	b	c
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	1	1	0
x_5	0	1	1
x_6	1	0	1
x_7	1	1	1



Zero sum subsets

x_1, x_2, x_4

x_2, x_3, x_5

x_1, x_3, x_6

x_1, x_2, x_3, x_7

Experimental analysis

Heuristic method implemented as a C program and tested on a standard PC with 2GB of main memory under Ubuntu Linux 8.10.

Experimental evaluation:

- on synthetic data, to assess **approximation ratio**
- on large real data, to assess **performances**

Experimental results - Synthetic genotypes

no. of genot.	no. of haplot.	no. of char.	avg. result	avg. ratio	no. of genot.	no. of haplot.	no. of char.	avg. result	avg. ratio
100	50	50	50	1	300	86	86	87	1.01
100	50	33	79.2	1.58	300	86	100	86	1
100	50	66	50	1	300	86	200	86	1
100	33	50	33	1	300	100	86	131.2	1.31
100	33	33	33.7	1.02	300	100	100	100.1	1
100	33	66	33	1	300	100	200	100	1
100	66	50	69.7	1.05	300	200	86	283	1.41
100	66	33	87.2	1.32	300	200	100	282.4	1.41
100	66	66	63	0.95	300	200	200	191	0.95
200	70	70	70.4	1	400	100	100	100.4	1
200	70	66	74.2	1.06	400	100	133	100	1
200	70	133	70	1	400	100	266	100	1
200	66	70	66	1	400	133	100	193.7	1.45
200	66	66	66	1	400	133	133	133	1
200	66	133	66	1	400	133	266	133	1
200	133	70	186.4	1.4	400	266	100	383.4	1.44
200	133	66	187.2	1.4	400	266	133	380.7	1.43
200	133	133	126	0.94	400	266	266	250	0.93